Option calibration of exponential Lévy models: Implementation and empirical results[*]

Jakob Söhl[†]   Mathias Trabs[‡]

February 28, 2012

Abstract

Observing prices of European put and call options, we calibrate exponential Lévy models nonparametrically. We discuss the implementation of the spectral estimation procedures for Lévy models of finite jump activity as well as for self–decomposable Lévy models and improve these methods. Confidence intervals are constructed for the estimators in the finite activity case. They allow inference on the behavior of the parameters when the option prices are observed in a sequence of trading days. We compare the performance of the procedures for finite and infinite jump activity based on real option data.

Keywords: European option · Jump diffusion · Self–decomposability · Confidence sets · Nonlinear inverse problem · Spectral cut–off

MSC (2010): 60G51 · 62G15 · 91B25

JEL Classification: C14 · G13

1 Introduction

Exponential Lévy models are frequently used for the purpose of pricing and hedging. They take jumps of the price process into account and they allow to model heavy tails in the returns appropriately. Moreover, they are capable of reproducing not only the volatility smile but also the fact that it becomes more pronounced for shorter maturities. While recovering these stylized facts, the structure of exponential Lévy models is at the same time easy enough to allow robust calibration procedures.

The calibration has mainly focused on parametric models, cf. Barndorff-Nielsen (1998); Eberlein, Keller, and Prause (1998); Carr, Geman, Madan, and Yor (2002) and the references therein. First nonparametric calibration procedures for finite activity Lévy models were proposed by Cont and Tankov (2004b) as well as Belomestny and Reiß (2006a). In these approaches no parametrization is assumed and thus the model misspecification is reduced. Recently, the spectral calibration procedure of the latter authors was studied further in two directions. The asymptotic confidence sets constructed by Söhl (2012) allow a more accurate statistical inference in the exponential Lévy model while the method of Trabs (2011) extends the spectral calibration to the infinite activity case, more precisely to self–decomposable Lévy processes.

The aim of this work is to improve the spectral calibration method and to apply it to real option data. In contrast to the method of Cont and Tankov (2004b) the spectral calibration is a

straight forward algorithm, where no minimization problem has to be solved. Therefore, already the originally proposed methods by Belomestny and Reiß (2006b) and by Trabs (2011) are quit fast owing to the FFT–algorithm. Whereas previous works focus on the asymptotic theory, we concentrate on the application of the method to realistic sample sizes. By a preprocessing of the data and a different choice of the weight functions we reduce the loss of the estimators significantly. These considerations are demonstrated by several simulations from the model by Merton (1976) and from the variance gamma model, introduced by Madan and Seneta (1990) and Madan, Carr, and Chang (1998).

Exponential Lévy models are studied in a wide range of pricing problems, for instance by Asmussen, Avram, and Pistorius (2004); Cont and Voltchkova (2005); Ivanov (2007) and the references therein. For their application, it is necessary to control the statistical errors in the calibration procedures, see Cont (2006). In general, there are two types of errors. The misspecification is the deviation from the model, while the calibration error is the deviation within the model. The approach of nonparametric calibration has the advantage that it reduces the error due to misspecification. The remaining calibration error can then be assessed by means of the confidence intervals. Nonparametric confidence intervals and bands for Lévy densities have been constructed by Figueroa-López (2011) based on high frequency observations. Söhl (2012) derived asymptotic confidence sets for the calibration of the risk neutral measure and based on observations of option prices and not on historical data. However, they turn out to be too conservative in simulations. To describe the behavior of the estimators more precisely, we construct confidence intervals based on a finite sample variance. As demonstrated by simulations these intervals perform well in terms of size and coverage probabilities.

We use real data of vanilla options on the German DAX index to compare the finite activity model to the self–decomposable one. Both models achieve a good calibration in the sense that the residuals between the given data and the calibrated model are small. In view of studies of Carr et al. (2002) and Aït-Sahalia and Jacod (2009), which indicate that a pure jump model should have a Blumenthal–Getoor index which is positive or even greater than one, this is surprising since the finite variation self–decomposable model has a Blumenthal–Getoor index equal to zero. Applying the calibration to a sequence of trading days, we obtain the evolution of the model parameters in time.

This paper is organized as follows: In Section 2 we state the model and the general estimation method. This is made precise for the finite activity case and the self–decomposable case in Sections 3 and 4, respectively. Simulations are contained therein. In Section 5 the confidence intervals are constructed and their performance is assessed in simulations. In Section 6 we apply the methods to real data and discuss our results. We conclude in Section 7. The appendix contains the proof of a corollary and a detailed documentation of our implementation.

2 Model and estimation principle

Denoting the riskless interest rate by $r \geq 0$ and the initial value by $S_0 > 0$, the price of a stock is given by

$$S_t = S_0 e^{rt + X_t}$$

where X_t is a Lévy process with characteristic triplet (σ^2, γ, ν) such that e^{X_t} is a martingale. Furthermore, we assume the jump part of X_t to be of finite variation. Therefore, $\mathbb{E}[e^{X_t}] = 1, t \geq 0$, implies the martingale condition

$$\frac{\sigma^2}{2} + \gamma + \int_{-\infty}^{\infty} (e^x - 1)\nu(\,\mathrm{d}x) = 0. \tag{1}$$

So far, nonparametric calibration methods exist in two different setups:

(FA) X_t has a finite activity, that is $\lambda := \|\nu\|_{L^1} < \infty$, with an absolutely continuous jump measure (Cont and Tankov, 2004b; Belomestny and Reiß, 2006a; Söhl, 2012).

(SD) X_t is self–decomposable with $\sigma = 0$. Especially, ν can be characterized by a k–function via $\nu(\mathrm{d}x) = k(x)/|x|\,\mathrm{d}x, x \in \mathbb{R} \setminus \{0\}$, where k is increasing on $\mathbb{R}_-$ and decreasing on $\mathbb{R}_+$. Additionally, $\alpha := k(0+) + k(0-) < \infty$ is assumed (Trabs, 2011).

Throughout we measure the time t in years. Typical parametric submodels of (FA) and (SD) are given by Examples 1 and 2, respectively. We will use them to study the performance of estimation methods in simulations.

Example 1 (Merton model). Merton (1976) introduced the first exponential Lévy model. Therein, the jumps are normally distributed with intensity $\lambda > 0$:

$$\nu(x) = \frac{\lambda}{\sqrt{2\pi v}} \exp\left(-\frac{(x - \eta)^2}{2v^2}\right), \quad x \in \mathbb{R}.$$

A realistic choice of the parameters is $\eta = -0.1$, $v = 0.2$ and $\lambda = 5$. Together with the volatility $\sigma = 0.1$ this determines the drift $\gamma = 0.379$ using the martingale condition (1).

Example 2 (Variance gamma model). Let (W_t) be a standard Brownian motion and (G_t) an independent Gamma process with mean rate one and variance rate ν that is $G_t \sim \Gamma(t/\nu, 1/\nu)$. Madan and Seneta (1990) defined the variance gamma process with parameters σ, ν and θ as the time changed Brownian motion with drift $X_t = \theta G_t + \sigma W_{G_t}, t \geq 0$. The characteristic function and the k–Function of (X_t) are given by

$$\varphi_t(u) = (1 + i\theta\nu u + \sigma^2\nu u^2/2)^{-t/\nu} \quad \text{and}$$

$$k_{VG}(x) = \frac{1}{\nu}e^{x/\eta_m}\mathbf{1}_{\{x<0\}}(x) + \frac{1}{\nu}e^{-x/\eta_p}\mathbf{1}_{\{x\geq 0\}}(x), \quad u, x \in \mathbb{R}$$

with $\eta_{p/m} := \sqrt{\theta^2\sigma^2/4 + \sigma^2\nu/2} \pm \theta\nu/2$, respectively. In our simulations we use the parameters $\sigma = 1.2, \nu = 0.2$ and $\theta = -0.15$. The value of $\gamma = 0.141$ is again given by the martingale condition. These choices imply $\alpha = k_{VG}(0+) + k_{VG}(0-) = 10$.

Let us fix a maturity $T > 0$, define the negative log–moneyness $x := \log(K/S_0) - rT$ and denote call and put prices by $\mathcal{C}(x, T) = S_0\mathbb{E}[(e^{X_T} - e^x)_+]$ and $\mathcal{P}(x, T) = S_0\mathbb{E}[(e^x - e^{X_T})_+]$, respectively. In terms of the option function

$$\mathcal{O}(x) := \begin{cases} S_0^{-1}\mathcal{C}(x, T), & x \geq 0, \\ S_0^{-1}\mathcal{P}(x, T), & x < 0, \end{cases}$$

our observations are given by

$$O_j = \mathcal{O}(x_j) + \delta_j\varepsilon_j, \quad j = 1, \ldots, N, \tag{2}$$

with noise levels $\delta_j > 0$ and independent, centered errors ε_j, satisfying $\mathrm{Var}(\varepsilon_j) = 1$ as well as $\sup_j \mathbb{E}[\varepsilon_j^4] < \infty$. The observation errors are due to the bid–ask spread and other market frictions. To estimate the Lévy triplet, we apply the Lévy–Khintchine representation of the characteristic function $\varphi_T(u) := \mathbb{E}[e^{iuX_T}]$. The pricing formula of Carr and Madan (1999)

$$\mathcal{F}\mathcal{O}(u) := \int_{-\infty}^{\infty} e^{iux}\mathcal{O}(x)\,\mathrm{d}x = \frac{1 - \varphi_T(u - i)}{u(u - i)}, \tag{3}$$

and the martingale condition (1) yield

$$\psi(u) := \frac{1}{T}\log(1 + iu(1 + iu)\mathcal{F}\mathcal{O}(u)) = -\frac{\sigma^2}{2}u^2 + i(\sigma^2 + \gamma)u + \int(e^{iux} - 1)e^x\nu(x)\,\mathrm{d}x. \tag{4}$$

Interpolating $(x_j, O_j)_{j=1,\ldots,N}$, we obtain empirical versions $\tilde{O}$ and $\tilde{\psi}$ of the option function and the characteristic exponent, respectively. While the theoretical results (Belomestny and Reiß, 2006a;

3

Trabs, 2011) concentrate on a linear interpolation of the observation, an additional smoothing by using B–splines of degree two might improve the estimators. In Section 3.2 we provide simulations with both interpolation methods to investigate the practical influence.

Given $\tilde{\psi}$, we can estimate the characteristics of the process from the spectral representation. Regularization of the procedure is achieved by cutting off frequencies larger than the regularization parameter $U > 0$. Since (FA) and (SD) need to be considered separately, the precise estimators are given in the Sections 3 and 4. Note that in both cases correction steps are necessary to satisfy the shape restriction of ν and the martingale condition (1) (see Appendix B.2 for details). If the latter one would be violated, the right–hand side of the pricing formula (3) could have a singularity at zero and thus we could not apply the inverse Fourier transform to obtain an option function from the calibration.

A critical question is the choice of the regularization parameter U. To circumvent the problem in simulations, we use an oracle cut–off value, that is the risk minimizing U. To calibrate real data, we employ a simple least squares approach. From theoretical consideration a penalty term, as used by Belomestny and Reiß (2006b), is necessary to avoid an over–fitting. Nevertheless, our practical experience with this method shows that the above mentioned correction steps, which are not included in the theory, lead to an auto–penalization: Owing to the Fourier techniques, a rigorous fit leads to high fluctuations of the estimator of the nonparametric part and thus the correction has a significant effect which in turn worsens the fit. Therefore, the least squares choice of tuning parameter works well at least for small noise levels. Additionally, an upper bound for the cut–off value excludes estimations with a too high variance (cf. Section 5).

3 The finite activity case

3.1 The estimators

In the (FA) framework we deduce from (4) the identity

$$\psi(u) = \frac{\sigma^2}{2}u^2 + i(\sigma^2 + \gamma)u + (\sigma^2/2 + \gamma - \lambda) + \mathcal{F}\mu(u) \quad \text{with } \mu(x) := e^x\nu(x).$$

The estimators of the parameters are defined by Belomestny and Reiß (2006a) as follows:

$$\hat{\sigma}^2 := \int_{-U}^{U} \text{Re}(\tilde{\psi}(u))w_\sigma^U(u)\mathrm{d}u, \tag{5}$$

$$\hat{\gamma} := -\hat{\sigma}^2 + \int_{-U}^{U} \text{Im}(\tilde{\psi}(u))w_\lambda^U(u)\mathrm{d}u, \tag{6}$$

$$\hat{\lambda} := \frac{\hat{\sigma}^2}{2} + \hat{\gamma} - \int_{-U}^{U} \text{Re}(\tilde{\psi}(u))w_\gamma^U(u)\mathrm{d}u, \tag{7}$$

with suitable weight functions w_σ^U, w_γ^U and w_λ^U. We propose to choose the weight functions differently to the weight functions used by Belomestny and Reiß (2006b). The idea is that the noise is particularly high in the high frequencies and thus it is desirable to assign less weight to the high frequencies. A smooth transition of the weight functions to zero at the cut off–value improves the numerical results significantly. Therefore, we would like the weight function and its first two derivatives to be zero at the cut–off value. With the side conditions on the weight functions this

leads to the following polynomials:

$$w_\sigma^U(u) := \frac{c_\sigma}{U^3}\left((2s+1)\left(\frac{u}{U}\right)^{2s} - 4(2s+3)\left(\frac{u}{U}\right)^{2s+2} + 6(2s+5)\left(\frac{u}{U}\right)^{2s+4}\right.$$
$$\left. - 4(2s+7)\left(\frac{u}{U}\right)^{2s+6} + (2s+9)\left(\frac{u}{U}\right)^{2s+8}\right),$$
$$w_\gamma^U(u) := \frac{c_\gamma}{U^2}\left(\left(\frac{u}{U}\right)^{2s+1} - 3\left(\frac{u}{U}\right)^{2s+3} + 3\left(\frac{u}{U}\right)^{2s+5} - \left(\frac{u}{U}\right)^{2s+7}\right),$$
$$w_\lambda^U(u) := \frac{c_\lambda}{U}\left((2s+3)\left(\frac{u}{U}\right)^{2s} - 4(2s+5)\left(\frac{u}{U}\right)^{2s+2} + 6(2s+7)\left(\frac{u}{U}\right)^{2s+4}\right.$$
$$\left. - 4(2s+9)\left(\frac{u}{U}\right)^{2s+6} + (2s+11)\left(\frac{u}{U}\right)^{2s+8}\right),$$

where all three functions are extended continuously by zero outside $[-U, U]$, c_σ, c_γ and c_λ are normalization constants and s reflects the a priori knowledge about the smoothness of ν. The gain of the new weight functions is discussed in Section 3.2. Using the shifted ψ

$$\psi_\nu(u) := \frac{1}{T}\log\left(1 - u(u+i)\mathcal{FO}(u+i)\right)$$

to estimate the jump density, we apply the idea from Belomestny and Reiß (2006b) owing to a better numerical behavior. Therefore, we define the estimator

$$\hat{\nu}(x) := \mathcal{F}^{-1}\left[\left(\tilde{\psi}_\nu(u) + \frac{\hat{\sigma}^2}{2}u^2 - i\hat{\gamma}u + \hat{\lambda}\right)w_\nu^U(u)\right](x) \tag{8}$$

with the empirical version $\tilde{\psi}_\nu$ of ψ_ν and a flat top kernel w_ν^U whose support is $[-U, U]$:

$$w_\nu^U(u) := w_\nu\left(\frac{u}{U}\right) \quad \text{with} \quad w_\nu(u) := \begin{cases} 1, & |u| \leq 0.05, \\ \exp\left(\frac{-\exp(-(|u|-0.05)^{-2})}{(|u|-1)^2}\right), & 0.05 < |u| < 1, \\ 0, & |u| \geq 1. \end{cases} \tag{9}$$

3.2 Simulations I

Let us first describe the setting of all of our simulations. In view of the higher concentration of European options at the money, the design points $\{x_1, \ldots, x_N\}$ are sampled deterministically from a normal distribution with mean zero and variance $1/2$. The observations O_j are computed from the characteristic function φ_T using the fast Fourier transform. The additive noise consists of independent, normal and centered random variables with variance $|\tau\mathcal{O}(x_j)|^2$ for some relative noise level $\tau > 0$. By choosing the sample size N and the deviation parameter τ, we determine the noise level of the observations. According to the existing theoretical results, it is well measured by the quantity

$$\varepsilon := \Delta^{3/2} + \Delta^{1/2}\|\delta\|_{l^\infty} \quad \text{with} \quad \Delta := \max_{j=2,\ldots,N}(x_j - x_{j-1}),$$

which takes the interpolation error and the stochastic error into account. The interest rate and time to maturity are set to $r = 0.06$ and $T = 0.25$, respectively.

Using the Merton model with the parameters of Example 1, we investigate the practical influence of two aspects of the procedure, which are mentioned above. The interpolation of the data (x_j, O_j) with linear B–splines is compared to the use of quadratic B–splines. The latter preprocessing is an additional smoothing of the data which achieves significant gains for higher noise levels. The other point of interest is the choice of the weight functions. Since it is known from the theory that the noise affects mainly the high frequencies, the polynomial weight functions greatly reduce the variance of the estimator. These improvements are illustrated in Figure 1: In the case of $\hat{\sigma}$ we calculate the RMSE from 500 Monte–Carlo iterations with and without quadratic splines and polynomial weight functions, respectively. This is done for different noise levels, whereby τ decreases from 0.03 to 0.015 and N increases from 50 to 400, simultaneously.
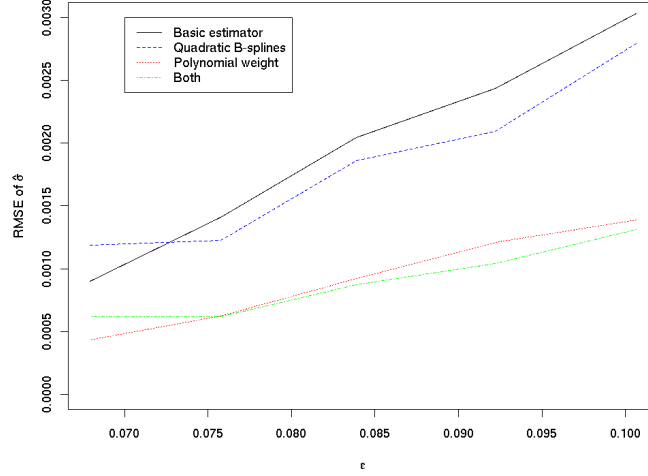
Figure 1: RMSE of $\hat{\sigma}$ for different noise levels with 500 Monte–Carlo iterations in each case. Usage of the linear and quadratic spline interpolation as well as usage of the weight function of Belomestny and Reiß (2006a) and the polynomial weight.

4 The self–decomposable framework

4.1 Construction of the estimators

Recall that $\sigma = 0$ is assumed in the (SD) setting. While the Blumenthal–Getoor index is zero in this case the parameter α describes the degree of activity of the process on a finer scale. The exponential scaled k–function $k_e(x) := \text{sgn}(x)e^x k(x), x \in \mathbb{R}$, which is the counterpart of μ in the (FA) case, will be the natural object to estimate. To calibrate the self–decomposable model, we need a different representation of ψ than before because of the infinite activity of these processes. Trabs (2011, Prop. 3.2) showed for $u \neq 0$

$$\psi(u) = D(u) + i\gamma u - \alpha \log(|u|) + \sum_{j=1}^{s-2} \frac{i^j (j-1)! \alpha_j}{u^j} + \rho(u) = i\gamma u + i \int_0^u \mathcal{F}k_e(v)\,\mathrm{d}v, \qquad (10)$$

where s is the smoothness of k away from zero, $\alpha_j := k^{(j)}(0+) + k^{(j)}(0-), j = 1, \ldots, s$, the function D is constant on the real half lines and the remainder ρ satisfies $\|u^{s-1}\rho(u)\|_\infty < \infty$. Owing to the polynomial decay of ρ, estimators of γ and α can be defined analogously to Section 3:

$$\hat{\gamma} := \int_{-U}^{U} \text{Im}(\tilde{\psi}(u)) w_\gamma^U(u)\,\mathrm{d}u,$$

$$\hat{\alpha} := \int_{-U}^{U} \text{Re}(\tilde{\psi}(u)) w_\alpha^U(u)\,\mathrm{d}u.$$

The weight functions are chosen such that they filter the coefficients of interest (cf. Trabs, 2011, Ass. 3). Choosing w_γ^U and w_α^U as polynomials, these integral conditions lead to a system of linear equations which determines the coefficients in the polynomials. Since the smoothness s is not known to the practitioner, the weights should satisfy the condition for some upper bound s_{max}.

6

In practice the choice $s_{max} = 6$ and

$$w_\gamma^U(u) = U^{-2} w_\gamma^1\left(\frac{u}{U}\right), \qquad w_\gamma^1(u) = \tfrac{135135}{1536}\left(-45u^5 + 560u^7 - 1890u^9 + 2376u^{11} - 1001u^{13}\right),$$

$$w_\alpha^U(u) = U^{-1} w_\alpha^1\left(\frac{u}{U}\right), \qquad w_\alpha^1(u) = \tfrac{1}{1024}\left(4729725u^6 - 56756700u^8 + 187297110u^{10}\right.$$
$$\left. - 231891660u^{12} + 96621525u^{14}\right)$$

works fine. The estimation of the nonparametric object k_e relies on the equation $\psi' = i\gamma + i\mathcal{F}k_e$, which follows from (10). Hence, we need an empirical version of ψ', too. Instead of the estimator proposed by Trabs (2011), we define

$$\hat{k}_e(x) := \begin{cases} \mathcal{F}^{-1}\big[(-\hat\gamma - i\tilde\psi'(u))\mathcal{F}W_k(u/U)\big](x), & x > 0, \\ \mathcal{F}^{-1}\big[(-\hat\gamma - i\tilde\psi'(u))\mathcal{F}W_k(-u/U)\big](x), & x < 0 \end{cases}$$

with a one–sided kernel function W_k that satisfies

Assumption 1. *We assume*

(i) $\operatorname{supp} W_k \subseteq [0, \infty)$,

(ii) $\int W_k = 1$, $\int x^l W_k(x)\,\mathrm{d}x = 0$ *for* $l = 1, \dots, s-1$ *and* $x^{2s-1}W_k(x) \in L^1(\mathbb{R})$,

(iii) $\int |u|^{2T\bar\alpha + 4}|\mathcal{F}W_k(u)| < \infty$ *for some upper bound* $\bar\alpha$ *of the true* α.

Since we know the position of the jump of k_e, the application of a one–side kernel function allows to estimate the k–function on the whole real line. The asymptotic analysis of Trabs (2011) carries over to this estimator as it is shown by the following corollary. Its proof is postponed to Appendix A. Recall the definitions of the Sobolev–type class $\mathcal{G}_s(R, \bar\alpha)$ and $\mathcal{H}_s(R, \bar\alpha)$ from Def. 4.1 and Def. 4.4 in Trabs (2011).

Corollary 1. *Assume* $s \geq 1, R, \bar\alpha > 0$ *and the conditions on* Δ *and* δ *from Thm. 4.5 in Trabs (2011). Furthermore, let* $\hat\gamma$ *be an estimator of* γ *satisfying* $\sup_{\mathcal{P}} \mathbb{E}_{\mathcal{P}}[|\hat\gamma - \gamma|^2]^{1/2} \lesssim \varepsilon^{(2s+1)/(2s+2T\bar\alpha+5)}$. *Using the cut–off value* $U := \varepsilon^{-2/(2s+2T\bar\alpha+5)}$, *we obtain*

$$\sup_{\mathcal{P} \in \mathcal{H}_s(R, \bar\alpha)} \mathbb{E}_{\mathcal{P}}[\|\hat{k}_e - k_e\|_{L^2}^2]^{1/2} \lesssim \varepsilon^{2s/(2s+2T\bar\alpha+5)}.$$

Remark. For $\bar\alpha \leq (5s-1)/(2T)$ the above defined estimator $\hat\gamma$ satisfies the assumed asymptotic risk bound of Thm. 4.2 in Trabs (2011) and thus the corollary, restricted to $\mathcal{P} \in \mathcal{G}_s(R, \bar\alpha) \cap \mathcal{H}_s(R, \bar\alpha)$, is applicable to the proposed procedure.

The condition (iii) in Assumption 1 is a smoothness condition on W_k. To be adaptive in α, we construct the kernel as $W_k(x) = P(x)w_\nu(2x - 1) \in C^\infty(\mathbb{R})$ with a polynomial $P(x)$ of degree $m + 1$. The coefficients of P are given by an $(m + 1)$ dimensional system of linear equations, defined by Property (ii), which can be solved numerically. Rearrangement of $\hat{k}_e$ ensures the necessary monotonicity of a k–function.

4.2   Simulations II

We simulate the variance gamma model from Example 2 and with the observation setting described in Section 3.2. The behavior of the proposed method for different noise regimes is shown in Table 1. As one expects from the theory, γ can be estimated better than α. In the realistic setting of 100 observations with one percent noise, the error of γ and α lay below three and seven percent, respectively. The one–sided kernel achieves good results for $\hat{k}_e(x)$ even for small $x \in \mathbb{R}$, indicated by Figure 2. However, there are still problems at zero. This emphasizes the importance of an separate estimator of α.

	N	50	100	100
	τ	0.01	0.01	0.05
RMSE	$\hat{\gamma}_{\text{(RMSE)}}$	0.0014	0.0004	0.0074
	$\hat{\alpha}_{\text{(RMSE)}}$	0.4196	0.6669	2.0231
RMISE	$\hat{k}_{e\ \text{(RMISE)}}$	0.5930	0.4980	0.4754

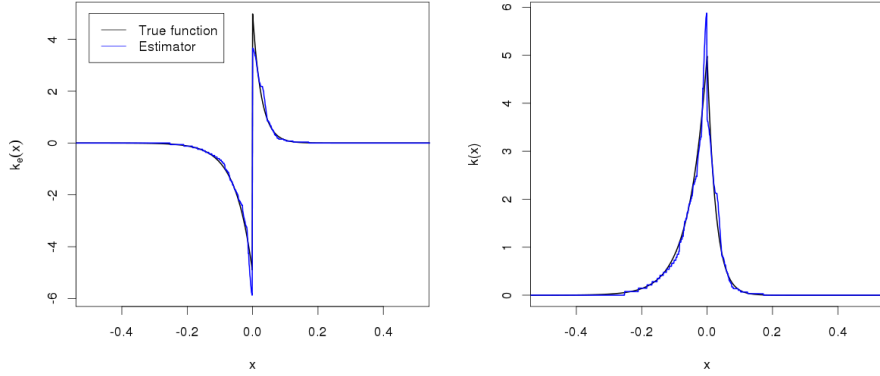Table 1: RMSE and RMISE using 1000 Monte–Carlo simulations of the variance gamma model.

Figure 2: Estimated functions k_e (*left*) and k (*right*) in a simulation of the variance gamma model with $N = 100$ and $\tau = 0.01$.

5   Confidence intervals

Söhl (2012) shows asymptotic normality of the estimators in the (FA) setup. These result may be used to construct confidence intervals. By (2.8) in Söhl (2012) it holds

$$\Delta\hat{\sigma}^2 := \hat{\sigma}^2 - \sigma^2 = \frac{2}{U^2}\int_0^1 \operatorname{Re}(\mathcal{F}\mu(Uu))w_\sigma^1(u)\mathrm{d}u + \frac{2}{U^2}\int_0^1 \operatorname{Re}(\Delta\tilde{\psi}(Uu))w_\sigma^1(u)\mathrm{d}u \qquad (11)$$

with $\Delta\tilde{\psi} = \tilde{\psi} - \psi$. The first term is the deterministic approximation error and the second term is the stochastic error. The choice of the cut–off value U allows a trade–off between these two errors. In order to construct confidence intervals, the cut–off value U is chosen such that the approximation error is asymptotically negligible. The term $\Delta\tilde{\psi}(Uu)$ is a logarithm, whose linearization at one we call $\mathcal{L}_{N,U}(u)$. We denote by $\mathcal{R}_{N,U}(u)$ the remainder of this linearization. The stochastic error may then be decomposed as

$$\frac{2}{U^2}\int_0^1 \operatorname{Re}(\Delta\tilde{\psi}(Uu))w_\sigma^1(u)\mathrm{d}u = \frac{2}{U^2}\int_0^1 \operatorname{Re}(\mathcal{L}_{N,U}(u))w_\sigma^1(u)\mathrm{d}u + \frac{2}{U^2}\int_0^1 \operatorname{Re}(\mathcal{R}_{N,U}(u))w_\sigma^1(u)\mathrm{d}u.$$

With an appropriate choice of the cut–off value the second part of the stochastic error is asymptotically negligible. We will call the first part in the decomposition the linearized stochastic error. Confidence intervals may be constructed in two different ways. One can derive confidence intervals either from the asymptotic variance or from the finite sample variance of the linearized stochastic errors. We will follow the second approach. Nevertheless, the confidence intervals are asymptotic in the sense that the remainder term of the stochastic errors and the approximation error are considered as negligible.

We assume that the noise levels of the observations (2) are given by the values $\delta_j = \delta(x_j)$, $j = 1, \ldots, N$, of some function $\delta : \mathbb{R} \to \mathbb{R}_+$. The observation points are assumed to be the quantiles $x_j = H^{-1}(j/(n+1))$, $j = 1, \ldots, N$, of a distribution with a c.d.f. $H : \mathbb{R} \to [0,1]$ and

p.d.f. h. For the definition of the confidence intervals we need the generalized noise level

$$\varrho(x) = \delta(x)/\sqrt{h(x)}, \tag{12}$$

which incorporates the noise of the observations as well as the density of the observations. Instead of assuming that the observation points are given by the quantiles of h one may also assume that the observation points are sampled randomly from the density h.

For σ^2 we calculate the finite sample variance $s_{\sigma^2}^2$ of the linearized stochastic errors using $\mathrm{Re}(z)^2 = \mathrm{Re}(|z|^2 + z^2)/2$ and (6.17) from Söhl (2012)

$$
\begin{aligned}
s_{\sigma^2}^2 &= \frac{4}{U^4}\mathbb{E}\left[\left(\int_0^1 \mathrm{Re}(\mathcal{L}_{N,U}(u))w_\sigma^1(u)\mathrm{d}u\right)^2\right] \\
&= \frac{2}{U^4}\mathrm{Re}\left(\mathbb{E}\left[\left|\int_0^1 \mathcal{L}_{N,U}(u)w_\sigma^1(u)\mathrm{d}u\right|^2 + \left(\int_0^1 \mathcal{L}_{N,U}(u)w_\sigma^1(u)\mathrm{d}u\right)^2\right]\right) \\
&= \frac{4\pi}{NT^2 e^{2T(\sigma^2/2+\gamma-\lambda)}}\left(\mathrm{Re}\int_0^1\int_0^1 f_U(u)\overline{f_U(v)}\mathcal{F}^{-1}(\varrho(y+\theta)^2)(U(u-v))e^{T\sigma^2 U^2(u^2+v^2)/2}\mathrm{d}u\mathrm{d}v\right. \\
&\qquad\left. + \mathrm{Re}\int_0^1\int_0^1 f_U(u)f_U(v)\mathcal{F}^{-1}(\varrho(y+\theta)^2)(U(u-v))e^{T\sigma^2 U^2(u^2+v^2)/2}\mathrm{d}u\mathrm{d}v\right), \tag{13}
\end{aligned}
$$

where $\theta = T(\sigma^2 + \gamma)$ and $f_U(u) = w_\sigma^1(u)(-u^2+iu/U)\exp(-T\mathcal{F}\mu(Uu))$. The confidence interval for λ is based on the similar finite sample variance of the corresponding linearized stochastic error. The only difference in the calculation for γ is that we use $\mathrm{Im}(z)^2 = \mathrm{Re}(|z|^2 - z^2)/2$. We observe that $\hat{\nu}(x)$ in (8) involves $\tilde{\psi}_\nu$ instead of $\tilde{\psi}$. Thus the confidence intervals for $\nu(x)$ are based on the linearization $\mathcal{L}_{N,U}^\nu(u)$ of $\Delta\tilde{\psi}_\nu = \tilde{\psi}_\nu - \psi_\nu$. The variance $s_{\nu(x)}^2$ of the linearized stochastic errors is

$$
\begin{aligned}
s_{\nu(x)}^2 &= \frac{U^2}{\pi^2}\mathbb{E}\left[\left(\int_0^1 \mathrm{Re}(\mathcal{L}_{N,U}^\nu(u)e^{-ixUu})w_\nu^1(u)\mathrm{d}u\right)^2\right] \\
&= \frac{U^2}{2\pi^2}\mathrm{Re}\left(\mathbb{E}\left[\left|\int_0^1 \mathcal{L}_{N,U}^\nu(u)e^{-ixUu}w_\nu^1(u)\mathrm{d}u\right|^2 + \left(\int_0^1 \mathcal{L}_{N,U}^\nu(u)e^{-ixUu}w_\nu^1(u)\mathrm{d}u\right)^2\right]\right) \\
&= \frac{U^6}{\pi NT^2 e^{-2T\lambda}}\left(\mathrm{Re}\int_0^1\int_0^1 g_U(u)\overline{g_U(v)}\mathcal{F}^{-1}(e^{-2y}\varrho(y+T\gamma)^2)(U(u-v))e^{T\sigma^2 U^2(u^2+v^2)/2}\mathrm{d}u\mathrm{d}v\right. \\
&\qquad\left. + \mathrm{Re}\int_0^1\int_0^1 g_U(u)g_U(v)\mathcal{F}^{-1}(e^{-2y}\varrho(y+T\gamma)^2)(U(u-v))e^{T\sigma^2 U^2(u^2+v^2)/2}\mathrm{d}u\mathrm{d}v\right), \tag{14}
\end{aligned}
$$

with $g_U(u) := w_\nu^1(u)(u^2+iu/U)\exp(-T\mathcal{F}\nu(Uu)-ixUu)$. The confidence interval for $\nu(0)$ is based on both the linearization $\mathcal{L}_{N,U}^\nu$ multiplied by the weight function w_ν^1 as well as the linearization $\mathcal{L}_{N,U}$ multiplied by $w_\sigma^1 \int_{-1}^1 v^2 w_\mu^1(v)\mathrm{d}v/2 - w_\lambda^1 \int_{-1}^1 w_\mu^1(v)\mathrm{d}v$, cf. the definition of w_0 by Söhl (2012). We denote by $\hat{s}_{\sigma^2}$, $\hat{s}_\gamma$, $\hat{s}_\lambda$, $\hat{s}_{\nu(x)}$ and $\hat{s}_{\nu(0)}$ the estimated standard deviations which are obtained by substituting σ, γ, λ, μ and ν by their respective estimators in (13), (14) and the analogous expressions. This yields feasible confidence intervals. For a level $\alpha > 0$ the $(1-\alpha)$–confidence intervals are then given by

$$
\begin{aligned}
I_{\sigma^2} &:= [\hat{\sigma}^2 - \hat{s}_{\sigma^2}q_{\alpha/2}, \hat{\sigma}^2 + \hat{s}_{\sigma^2}q_{\alpha/2}], \\
I_\gamma &:= [\hat{\gamma} - \hat{s}_\gamma q_{\alpha/2}, \hat{\gamma} + \hat{s}_\gamma q_{\alpha/2}] \\
I_\lambda &:= [\hat{\lambda} - \hat{s}_\lambda q_{\alpha/2}, \hat{\lambda} + \hat{s}_\lambda q_{\alpha/2}], \\
I_{\nu(x)} &:= [\hat{\nu}(x) - \hat{s}_{\nu(x)}q_{\alpha/2}, \hat{\nu}(x) + \hat{s}_{\nu(x)}q_{\alpha/2}], \\
I_{\nu(0)} &:= [\hat{\nu}(x) - \hat{s}_{\nu(0)}q_{\alpha/2}, \hat{\nu}(x) + \hat{s}_{\nu(0)}q_{\alpha/2}],
\end{aligned} \tag{15}
$$

where q_α denotes the $(1-\alpha)$–quantile of the standard normal distribution.
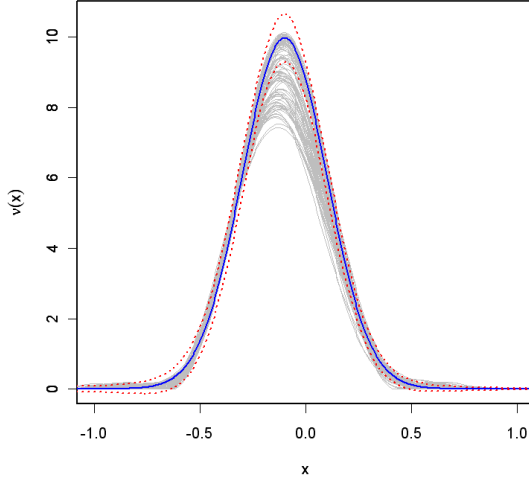
Figure 3: True Lévy density *(blue, solid)*, pointwise 95% confidence intervals *(red, dotted)* and 100 estimated Lévy densities *(grey)* from a Monte Carlo simulation.
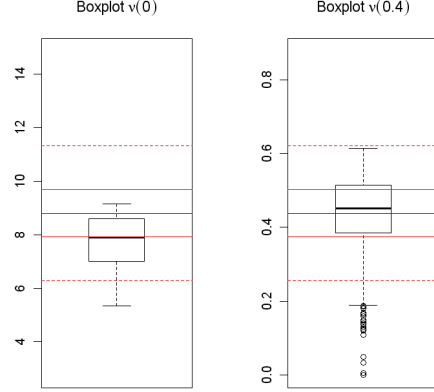
Figure 4: Boxplots of $\hat{\nu}(0)$ and $\hat{\nu}(0.4)$ from 1000 Monte Carlo iterations with true value *(blue)*, theoretical quantiles of the linearized stochastic error for 25% and 75% *(red, solid)* as well as for 2.5% and 97.5% *(red, dashed)*.

We examine the performance of the confidence intervals by simulations from the Merton model with parameters as in Example 1, with interest rate $r = 0.06$ and with maturity $T = 0.25$ as in Section 3.2. We sample $N = 100$ strike prices deterministically and take the relative noise level to be $\tau = 0.01$. To coincide with the theory, we interpolate the corresponding European call prices linearly and use the weight functions by Belomestny and Reiß (2006b) with smoothness parameter $s = 2$. In the real data application in Section 6 take advantage of the above improvements. Upon fixing an upper bound of 28 for the cut–off values, we perform 1000 Monte Carlo iterations.

Figure 3 illustrates the true Lévy density, pointwise 95% confidence intervals and the first one hundred estimated Lévy densities from the Monte Carlo simulation. The graph shows that the confidence intervals describe well the deviation of the estimated jump densities. The negative bias around zero might come from the smoothing which naturally tends to smooth out peaks, cf. (Härdle, 1990, Chap. 5.3). The confidence interval $I_{\nu(0)}$ is larger than indicated in Figure 3. By the smoothness of the estimated curves the larger variance of $\hat{\nu}(0)$ leads to an increased variance of $\hat{\nu}(x)$ in a neighborhood of zero. This is the second reason why deviation of the estimated curves is larger at zero. Figure 4 shows boxplots of $\hat{\nu}(0)$ and $\hat{\nu}(0.4)$, the true value, the theoretical 25% and 75% quantiles as well as the theoretical 2.5% and 97.2% quantiles based on the linearized stochastic errors. We see that $\hat{\nu}(0)$ has a negative bias and that the distribution of $\hat{\nu}(0.4)$ fits well to the theoretical quantiles of the linearized stochastic errors.

We asses the performance of the confidence intervals (15) with levels $\alpha = 0.5$ and $\alpha = 0.05$ for the parameters σ^2, γ, λ, $\nu(0)$ and $\nu(0.4)$ in a Monte Carlo simulation with 1000 iterations. We approximate the coverage probabilities of the confidence sets by the percentage of confidence intervals which contain the true value. Table 2 gives the approximate coverage probabilities from the Monte Carlo simulation.

	σ^2	γ	λ	$\nu(0)$	$\nu(0.4)$
$\alpha = 0.5$	54%	47%	50%	49%	47%
$\alpha = 0.05$	97%	95%	91%	98%	95%

Table 2: Approximate coverage probabilities of $(1 - \alpha)$–confidence intervals from a Monte Carlo simulation with 1000 iterations.
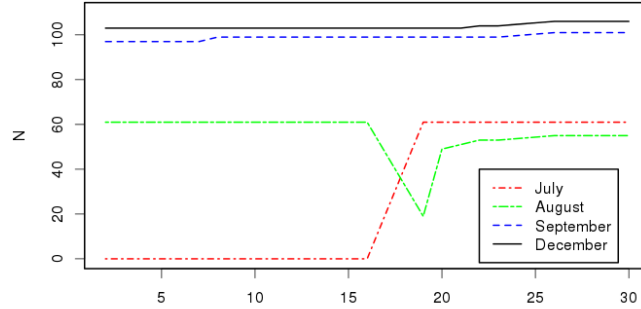

Figure 5: Number of observed prices of put and call options during May ,2008.

6   Empirical study

We apply the calibration methods to a data set from the Deutsche Börse database Eurex[1]. It consists of settlement prices of European put and call options on the DAX index from May 2008. Therefore, the prices are observed before the latest financial crises and thus the market activity is stable. The interest rate r is chosen for each maturity separately according to the put–call parity at the respective strike prices. The time to maturity T is measured in years. The number of our observations is given in Figure 5.

6.1   Comparison of (FA) and (SD)

Let us first focus on one (arbitrarily chosen) day. Hence, we calibrate the option prices of May 29, 2008, with all four different maturities to both, the (FA) and the (SD) setting. The results are summarized in Table 3 and Figure 6. Using the complete estimation of the models, we generate the corresponding option functions $\hat{\mathcal{O}}$. They are graphically compared to the given data points and we calculate the residual sum of squares $RSS = \sum_j (O_j - \hat{\mathcal{O}}(x_j))^2$. For all maturities both methods yield good fits to the data. However, for longer maturities, especially the calibration of options with four months to maturity, minor problems occur in the (SD) calibration. The calibration at other trading days confirms this weakness of the (SD) method for larger T. This coincides with the asymptotic analysis of Trabs (2011) and Corollary 1 where longer maturities lead to slower convergence rates of the risk.

Moreover, Figure 6 shows that the estimated option function $\hat{\mathcal{O}}$ which results from the (SD) calibration does not exactly recover the peak of $\mathcal{O}$. In all maturities and in both models the Lévy density has more weight on the negative half line and thus there are more negative jumps than positive ones priced into the options. This coincides with the empirical findings in the literature (see eg, Cont and Tankov, 2004a).

In Table 3 we see a tendency that higher values of $\hat{\sigma}$ in (FA) correspond to higher $\hat{\alpha}$ in the (SD) model. The scatter plot in Figure 7 displays the pairs $(\hat{\sigma}, \hat{\alpha})$ in the estimations of all trading

---

[1]provided through the SFB 649 "Economic Risk"

		61	55	101	106
	N	61	55	101	106
	T	0.136	0.233	0.311	0.564
(FA)	$\hat{\sigma}$	0.109	0.111	0.106	0.097
	$\hat{\gamma}$	0.224	0.163	0.176	0.193
	$\hat{\lambda}$	3.498	1.716	1.862	2.314
	$\sqrt{RSS}$	0.003	0.007	0.005	0.007
(SD)	$\hat{\gamma}$	0.337	0.297	0.378	0.227
	$\hat{\alpha}$	25.313	30.084	36.123	16.557
	$\sqrt{RSS}$	0.006	0.006	0.021	0.014

Table 3: Results of the estimations from option prices from May 29, 2008.

days with maturity in July or August. Note that there are no more than 60 observations for each point and thus the stochastic error is relatively big. Nevertheless, the positive correlation between σ and α is confirmed weakly and it is in line with the expectation that fluctuations of the stock are modeled by the diffusion part in the former model and by the infinitely high activity of the jump part in the latter one. The other way around, this relation shows that α is indeed a meaningful measure of the jump activity in the (SD) setting.

6.2 (FA) across trading days

The aim of this section is twofold. By considering more than one day we investigate the stability of the (FA) estimation procedure. Moreover, calibrating the model across the trading days in May, 2008, shows the development of the model along the time line and with small changes in the maturities. To profit from the higher observation number, we apply the calibration procedure for the (FA) case to the options with maturity in September and December.

To apply the confidence intervals (15) of Section 5, we need the noise function ϱ from (12). By a rule of thumb we assume δ to be 1% of the observed prices $\mathcal{O}(x_j)$ (cf. Cont and Tankov, 2004a, p. 439), for $j = 1, \ldots, N$. The density h of the strikes is estimated by a triangular kernel estimator, where the bandwidth is chosen by Silverman's rule of thumb.

The estimations of the parameters are displayed in Figure 8. Furthermore, the 95% confidence intervals for the December options are shown. The estimated volatility $\hat{\sigma}$ fluctuates between 0.1 and 0.12. The confidence sets imply that there is no significant difference of the two maturities. Both $\hat{\gamma}$ and $\hat{\lambda}$ decrease for higher durations: On the one hand the curves of December lay significantly below the ones of September, on the other hand the graphs have a slight positive trend with respect to the time axis, which means with smaller time to maturity.

Figure 9 displays the estimated jump densities. All jump measures have a similar shape which is in line with real data calibration of Belomestny and Reiß (2006b). In contrast to Cont and Tankov (2004b) the densities are unimodal or have only minor additional modes in the tails, which may be artefacts of the spectral calibration method. The tails of $\hat{\nu}$ do not differ significantly, while the different heights reflect the development of the jump activities $\hat{\lambda}$. Showing the point–wise confidence sets around $\hat{\nu}(x)$ on a fine grid, Figure 10 confirms this suggestions. There is an obvious trend to small negative jumps in all data sets which is in line with the stylized facts of option pricing models.

7 Conclusions

To reduce the model misspecification it is reasonable to use a nonparametric model for option pricing. However, the nonlinear inverse problem, which occurs by calibrating the model, is more difficult to solve than parametric calibration problems and needs non–standard algorithms. We could improve the existing spectral calibration procedures for the finite activity (FA) Lévy model and the self–decomposable (SD) Lévy model. Owing to the fast Fourier transform, the method
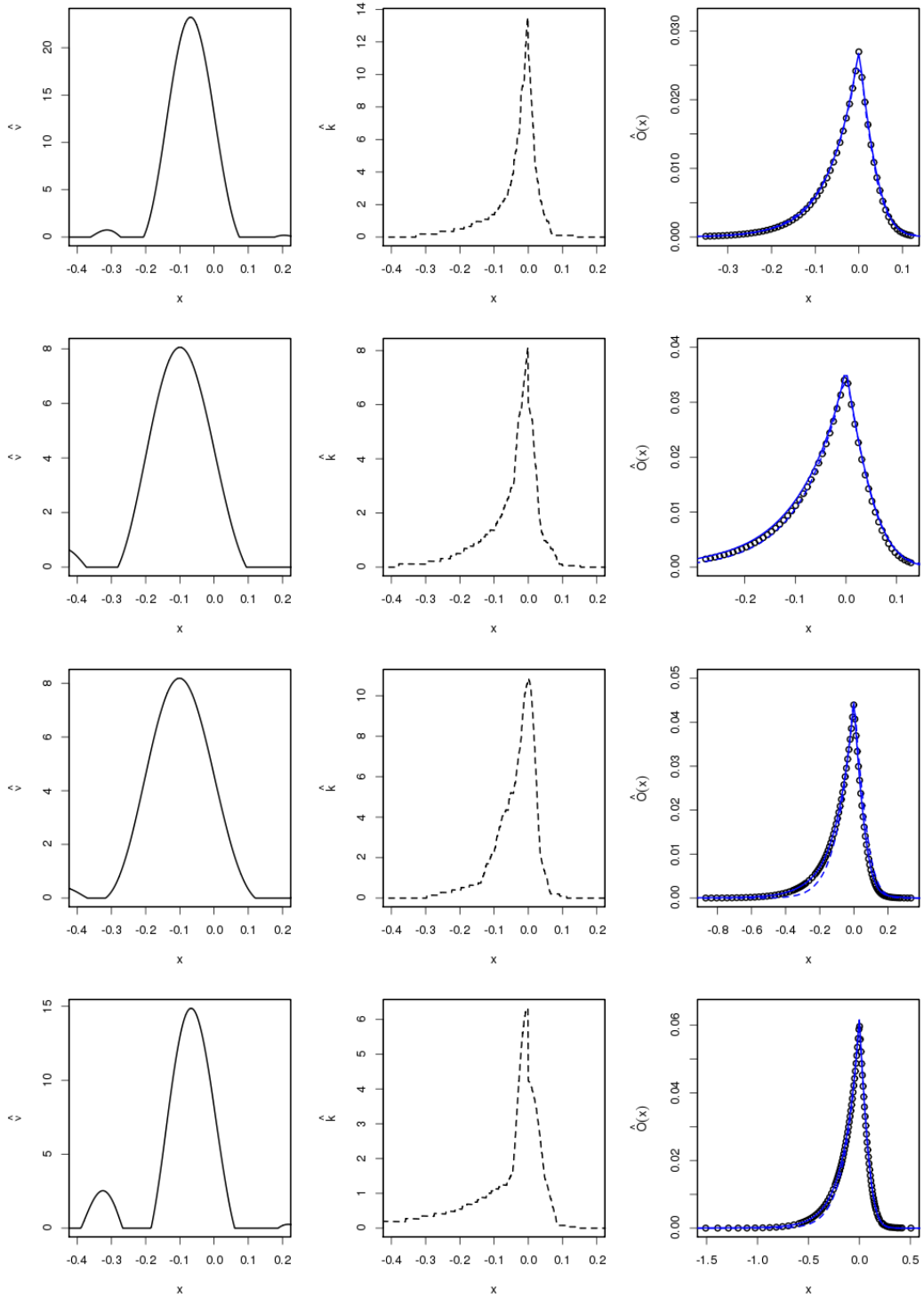
Figure 6: Estimated jump density *(left)*, k–function *(center)* as well as calibrated option functions in the (FA) *(right, solid)* and (SD) *(right, dashed)* setting and given data from May 29, 2008 *(right, points)*.
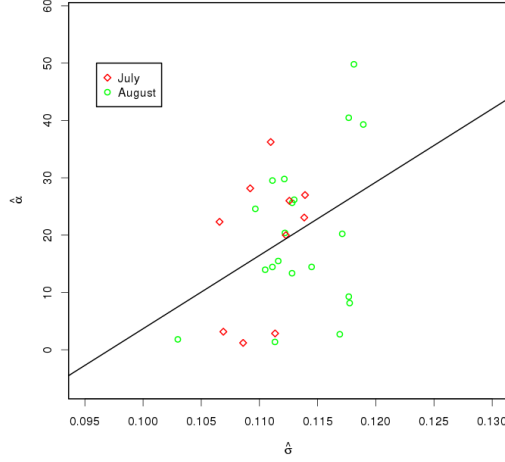
13

Figure 7: Pairs $(\hat{\sigma}, \hat{\alpha})$ estimated from options from each market day in May, 2008, with Maturity in July *(red diamonds)* and August *(green circles)* and regression line *(black)*.

is computationally fast and admits convincing results in simulations and real data applications. Determining the finite sample variances of the linearized estimators, we obtain confidence sets, which allow a precise analysis of the noise of the estimators.

Our empirical investigations show that both models can be calibrated well to European option prices. However, (FA) is more suitable for longer maturities. Using the derived confidence sets, we can observe significant changes of the (FA) model over time. While the volatility has no systematic trend, the jump activities decrease for longer maturities and thus the Lévy densities become flatter.

To avoid misspecification of the model, we are convinced that the nonparametric approach should be pushed forward theoretically and in practice, in particular, in view the high number of available observations in highly liquid markets. Of further interest would be extensions of the method to models whose jump part is not of finite variation as well as the application to hedging and risk management problems.

A   Proof of Corollary 1

W.l.o.g. it suffices to estimate the risk on $\mathbb{R}_+$. We decompose the risk to

$$
\mathbb{E}_{\mathcal{P}}[\|\hat{k}_e - k_e\|_{L^2(\mathbb{R}_+)}^2] = \mathbb{E}_{\mathcal{P}}\Big[\Big\|\mathcal{F}^{-1}\Big((-\hat{\gamma} - i\tilde{\psi}'(u))\mathcal{F}W_k(\frac{u}{U})\Big) - k_e\Big\|_{L^2(\mathbb{R}_+)}^2\Big]
$$

$$
\leq 3\int_{\mathbb{R}_+}\Big|\big(k_e * (UW_k(U\bullet))(x) - k_e(x)\big|^2\,\mathrm{d}x + 3\mathbb{E}_{\mathcal{P}}[|\gamma - \hat{\gamma}|^2]\int_{\mathbb{R}_+}|UW_k(Ux)|^2\,\mathrm{d}x
$$

$$
+ 3\mathbb{E}\Big[\int_{\mathbb{R}_+}\Big|\mathcal{F}^{-1}\big((\tilde{\psi}'(u) - \psi'(u))\mathcal{F}W_k(\frac{u}{U})\big)(x)\Big|^2\,\mathrm{d}x\Big]
$$

$$
=:D + G + S.
$$

The deterministic error term D can be bounded exactly as in the proof of (Trabs, 2011, Thm 4.5) with $\tau = 1$ and use of the properties of the kernel W_k. The assumption on $\hat{\gamma}$ and the choice of U yield $G \lesssim U\varepsilon^{(4s+2)/(2s+2T\bar{\alpha}+5)} = \varepsilon^{4s/(2s+2T\bar{\alpha}+5)}$. It remains to estimate the stochastic error term S. Applying Plancherel's equality, the bound of $|\tilde{\psi}' - \psi'|$ from (Trabs, 2011, p. 27) and the decay
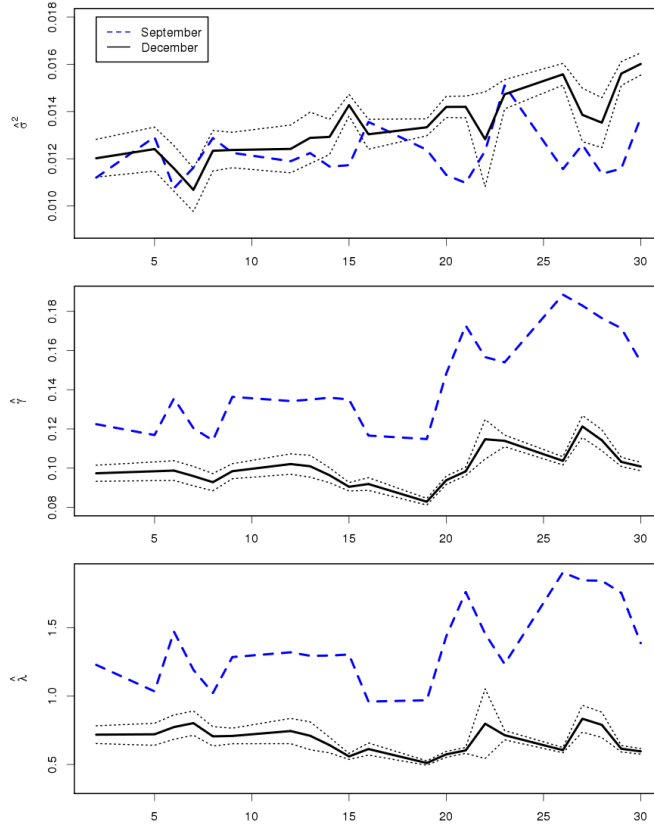
Figure 8: At each market day in May, 2008, estimated σ^2 *(top)*, γ *(center)* and λ *(bottom)* from options with maturities in September *(dashed)* and December *(solid)* and confidence intervals *(dotted)* for the latter ones.
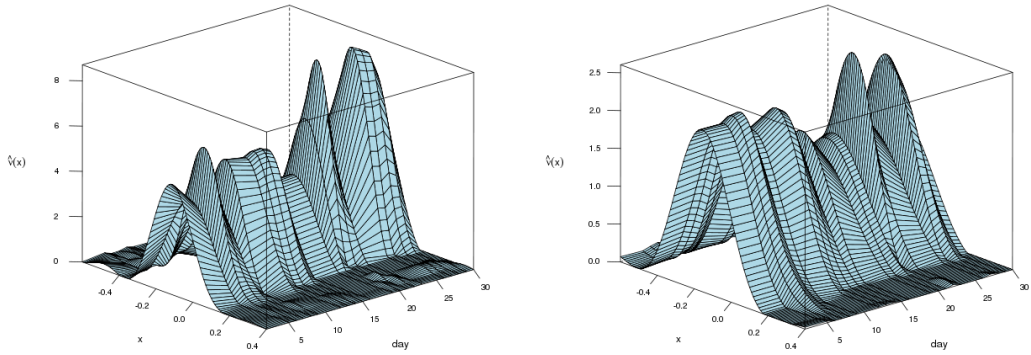

Figure 9: Estimation of ν for maturity in September *(left)* and December *(right)*.
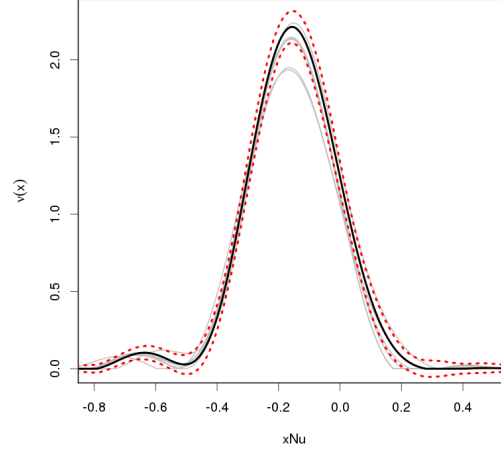
Figure 10: Empirical confidence sets *(dotted)* of $\hat{\nu}(x)$ *(solid)* in the estimation from option prices of May 6, 2008, with maturity in December and estimated jump densities from May 2 to 12 with the same maturity *(grey)*.

of $\mathcal{F}W_k$, we obtain

$$
\begin{aligned}
S &\leq \mathbb{E}_{\mathcal{P}}\big[\big\|\big(\tilde{\psi}'(u) - \psi'(u)\big)\mathcal{F}W_k(\tfrac{u}{U})\big\|_{L^2}^2\big] \\
&\lesssim \int_{\mathbb{R}} \frac{u^4}{|\kappa(u)|^2}\Big(\mathbb{E}\big[|\mathcal{F}(\tilde{\mathcal{O}} - \mathcal{O})(u)|^2\big] + \mathbb{E}\big[|\mathcal{F}\big(x(\tilde{\mathcal{O}} - \mathcal{O})(x)\big)(u)|^2\big]\Big)\big|\mathcal{F}W_k(\tfrac{u}{U})\big|^2\,\mathrm{d}u \\
&\lesssim (\Delta^4 + \Delta\|\delta\|_\infty^2)\int_{\mathbb{R}} u^{2T\bar{\alpha}+4}\big|\mathcal{F}W_k(\tfrac{u}{U})\big|^2\,\mathrm{d}u \\
&\lesssim \varepsilon^2 U^{2T\bar{\alpha}+5} = \varepsilon^{4s/(2s+2T\bar{\alpha}+5)}.
\end{aligned}
$$

B   Implementation in R

B.1   Global parameters

We give here a detailed description of our implementation of the calibration method for the finite activity case. Some global parameters to adjust the code and some auxiliary functions are necessary. These are namely `Zeta` needed for simulations, `CFHat` and `ZetaHat` to reproduce observations from the estimated model, the Fourier transform function `FT` and a continuous logarithm `logc`. The calibration itself is done in the function `calibration` documented in Section B.2. We use these functions either to calibrate simulations or to estimate real data in Section B.3.

There are a few parameters to control the execution of the program:

- `model` permits the values "Merton", "Kou" and "real data". In the first two cases observations will be simulated with the corresponding model. Picking the last value, the user has to supply the necessary data in the workspace of R before running the script (vector of strikes `k`, vector of option prices `op`, interest rate `r` and maturity `T`).

- In case of simulations the user has to specify `noiseLevel`, the `sampleSize` and the number of Monte Carlo iterations in `monteCarlo`. Furthermore, `design` decides whether the observed strikes are sampled "deterministic" or "random".

16

- The choice of the cut–off parameter U is adjusted by `mode`. In Belomestny and Reiß (2006b) are three possibilities proposed which are (in modifications) also available in this implementation. Possible values for `mode` are:

  – "oracle": U^* and U^*_ν minimize the discrepancy between the estimators and the given values, separately for $(\sigma^2, \gamma, \lambda)$ and ν.

  – "flat": The cut–offs correspond to points where the estimators stabilize:

  $$U^* = \mathrm{argmin}_U \left(\left| \frac{d}{dU} \hat{\sigma}_U \right| + \alpha U \right), \quad \alpha > 0,$$

  $$U^*_\nu = \mathrm{argmin}_{U_\nu \leq U^*} \left\| \frac{d}{dU} \hat{\nu}_{U_\nu} \right\|_{L^2}.$$

  In our simulations and real data estimations we used $\alpha = 10^{-5}$.

  – "PLS": The common cut–off U^* solves

  $$\inf_U \left[\sum_{i=1}^N |C(K_i; \mathcal{T}_U) - Y_i|^2 + \alpha \int_{\mathbb{R}} |\hat{\nu}''_U(x)|^2 \, dx \right], \quad \alpha > 0,$$

  where $C(K; \mathcal{T}_U)$ is the price at strike K computed using Levy triplet $\mathcal{T}_U = (\hat{\sigma}_U, \hat{\gamma}_U, \hat{\nu}_U,)$. In our simulations and real data estimations we use $\alpha = 0$.

  – "fix": The same cut–off values `Ufix` and `UfixNu` are used to estimate the parameters and the jump density, respectively.

- If the variable `linear` is true then the observation will be interpolated linearly. Otherwise quadratic B–splines are used.

- The smoothness of the jump density is set in parameter `s` (in the context of Belomestny and Reiß (2006a) `s` implies a smoothness $s = 2*$`s` of ν).

We remark that the library `cobs` is necessary to use this program. It provides the spline interpolation.

B.2 The function `calibration`

Given a vector of log–strikes `sk` and corresponding call option prices `snop` this function performs the calibration of the model and returns the cut–off values U, U_ν, the model parameters $\hat{\sigma}, \hat{\gamma}, \hat{\lambda}$ and the estimated jump density $\hat{\nu}$ on a grid `x`. If a simulation is performed then also the quadratic L^2–error of $\hat{\nu}$ is given back.

```
calibration<-function(sk, snop){
  ...
  # return
  list(U=U, UNu=UNu, sigmaHat=sigmaHat, gammaHat=gammaHat, lambdaHat=
    lambdaHat, x=x, nuHat=nuHat, nuError2=nuError2)
}
```

To approximate the $\mathcal{O}$–function from the given discrete points (`sk`, `snop`), we switch from the log–scale to the normal one, because we can use the convexity of the function $K \mapsto C(K, T) = e^{-rT}\mathbb{E}[(S_T - K)^+]$ for the quadratic spline interpolation. Furthermore, we extrapolate the observations by adding boundary points whose position is set by `extrapolation`. To get an option value for the new strike near zero we use the slope of $C(K, T)$ in K at 0. In the case of quadratic spline interpolation the function `cobs` from package `cobs` is used to interpolate and evaluate the result on the logarithmic scale on a fine grid. We get the points (`knew`, `opnew`) and use the put–call parity to have a discrete version of the function $k \mapsto \tilde{\mathcal{O}}(k - rT)$.

```
extrapolation<−0.005

extraK<−rep(0,length(sK)+2)
extraK[1]<−extrapolation*sK[1]
extraK[2:(length(sK)+1)]<−sK
extraK[length(sK)+2]<−sK[length(sK)]*1/extrapolation

extraOp<−rep(0,length(sK)+2)
extraOp[1]<−1−exp(−r*T)*extraK[1]
extraOp[2:(length(sK)+1)]<−snop
extraOp[length(sK)+2]<−0

BI<−log(extraK[1])
BE<−log(extraK[length(extraK)])
knew<−seq(BI,BE,length=2^12)

if(linear){
  OpofK <− approxfun(extraK,extraOp, method="linear",rule=2,ties=
      mean)
  opnew <− OpofK(exp(knew))
}else{
  css<−cobs(extraK,extraOp,constraint="convex",nknots=min(100,length
      (sK)−2),degree=2)
  opnew<−predict(css,exp(knew))[,2]
}

opnew<−opnew−pmax(0.,1−exp(knew−r*T))
opnew<−pmax(0,opnew)
```

We can now apply the Fourier transform to get first the estimate of $v \mapsto \tilde{\varphi}_T(v-i) = 1 + iv(1 + iv)\mathcal{F}\tilde{\mathcal{O}}(v)$ and calculate then $\tilde{\psi}(v) = 1/T\log(\tilde{\varphi}(v-i))$ by profiting from symmetry. Now we have (v, `psi`) and define `L` as the index where `v` has value 0.

```
z<−FT(knew−r*T,opnew,TRUE)  #x=k−rT,  z=FO(v)

v<−z$u
phi<−z$fy*1i*v*(1+1i*v)+1

psi<−rep(0,length(phi))
psi[(length(v)/2):length(v)]<−1/T*logc(phi[(length(v)/2):length(v)])
psi[1:(length(v)/2−1)]<−Re(psi[(length(v)−1):(length(v)/2+1)])−1i*Im
    (psi[(length(v)−1):(length(v)/2+1)])

L<−length(v)/2
```

In the following we calculate for numerous cut–off values U the estimators $\hat{\sigma}^2, \hat{\gamma}$ and $\hat{\lambda}$ and save each in `sigma2HatCur`, `gammaHatCur` and `lambdaHatCur`, respectively. The cut–off values are multiples of the mesh of `v` and their number is given by `cutOffIterations`. Since the calculation time in the "PLS"–mode is much longer than in the other modes but the picked cut–off is smaller in general we choose less iterations in "PLS". If the mode is "fix", only one iteration with the fixed cut–off suffices. Because of a large bias for small U we start with `(v[2]-v[1])*11` and therefore end at `(v[2]-v[1])*(10+cutOffIterations)`. Later we need `bestError` for the decision in the oracle–mode. With use of symmetry it is enough to calculate (`v_cut`, `psi_cut`) on the positive half–axis.

```
if(mode=="oracle" | mode=="flat")
```

```
      cutOffIterations<-130
   else if(mode=="PLS")
      cutOffIterations<-90
   else if(mode=="fix")
      cutOffIterations<-1

   sigma2HatCur<-rep(0,cutOffIterations)
   gammaHatCur<-rep(0,cutOffIterations)
   lambdaHatCur<-rep(0,cutOffIterations)
   bestError<--1
   for(i in 1:cutOffIterations){
      if(mode=="fix"){
         UCur<-Ufix
         gridU<-round(Ufix/(v[2]-v[1]))
      }else{
         UCur <- (v[2]-v[1])*(i+10)
         gridU <- i+10
      }
      v_cut<-v[L:(L+gridU)]
      psi_cut<-psi[L:(L+gridU)]
      ...
   }
```

Within this loop we approximate the integrals in (5) - (7) with a composite trapezoidal rule and use the polynomial weight functions from above.

```
   w<-rep(1,length(psi_cut))
   w[length(psi_cut)]<-0.5

   wFkt<-function(x){(2*s+1)*x^(2*s)-4*(2*s+3)*x^(2*s+2)+6*(2*s+5)*x
       ^(2*s+4)-4*(2*s+7)*x^(2*s+6)+(2*s+9)*x^(2*s+8)}
   weight<-wFkt(v_cut/UCur)
   weight[length(weight)] <- weight[length(weight)]-2*sum(w*weight)
   sigma2HatCur[i]<-sum(w*weight*Re(psi_cut))
   sigma2HatCur[i]<--2*sigma2HatCur[i]/sum(w*abs(v_cut)^{2}*weight)

   wFkt<-function(x){x^(2*s+1)-3*x^(2*s+3)+3*x^(2*s+5)-x^(2*s+7)}
   weight<-wFkt(v_cut/UCur)
   gammaHatCur[i]<-sum(w*weight*Im(psi_cut))
   gammaHatCur[i]<-gammaHatCur[i]/sum(w*weight*v_cut)-sigma2HatCur[i]

   wFkt<-function(x){(2*s+3)*x^(2*s)-4*(2*s+5)*x^(2*s+2)+6*(2*s+7)*x
       ^(2*s+4)-4*(2*s+9)*x^(2*s+6)+(2*s+11)*x^(2*s+8)}
   weight<-wFkt(v_cut/UCur)
   weight[length(weight)] <- weight[length(weight)]-2*sum(w*weight*v_
       cut^2)/v[L+gridU]^2
   lambdaHatCur[i]<-sum(w*Re(psi_cut)*weight)
   lambdaHatCur[i]<--lambdaHatCur[i]/sum(w*weight)+gammaHatCur[i]+
       sigma2HatCur[i]/2
```

The choice of U depends on the mode. In "oracle" in every iteration step we calculate the distance between the estimators and the true values as an error term and select the minimizing U. In the case of "fix" the cut–off is unique.

```
   if(mode=="oracle"){
```

```r
      error<-1*abs(sigma2HatCur[i]-sigma^2)+1*abs(gammaHatCur[i]-gamma
          )+1*abs(lambdaHatCur[i]-lambda)
      if((i==1) | error<bestError){
        U<-UCur
        sigmaHat<-sqrt(pmax(0,sigma2HatCur[i]))
        gammaHat<-gammaHatCur[i]
        lambdaHat<-lambdaHatCur[i]
        bestError<-error
      }
    }
    if(mode=="fix"){
      U<-UCur
      sigmaHat<-sqrt(pmax(0,sigma2HatCur[i]))
      gammaHat<-gammaHatCur[i]
      lambdaHat<-lambdaHatCur[i]
      bestError<-error
    }
```

After the loop and in case of the "flat"–mode we calculate a moving average of order two of the differences in the discrete curve $U \mapsto \hat{\sigma}_U$ and select the minimum of these values with respect to a penalty for big U.

```r
  if(mode=="flat"){
    diff<-diff(sigma2HatCur[1:length(sigma2HatCur)])
    ma<-rep(0,length(diff))
    maOrder<-2
    for(i in (1:length(diff))){
      ma[i]<-sum(abs(diff[max(i-maOrder,1):min(i+maOrder-1,length(diff
          ))]))/(min(i+maOrder,length(diff))-max(i-maOrder,1))
    }

    Us<-(v[2]-v[1])*(10+(1:(cutOffIterations-1)))
    alpha<-1e-5
    i<-which.min(ma+alpha*Us)
    U<-Us[i]
    sigmaHat<-sqrt(pmax(0,sigma2HatCur[i]))
    gammaHat<-gammaHatCur[i]
    lambdaHat<-lambdaHatCur[i]
  }
```

To estimate ν as in (8) we first have to build $\tilde{\psi}_\nu$:

```r
  z<-FT(knew-r*T,exp(-knew+r*T)*opnew,TRUE)
  phiNu<-1-v*(v+1i)*z$fy
  psiNu<-rep(0,length(phiNu))
  psiNu[(length(v)/2):length(v)]<-1/T*logc(phiNu[(length(v)/2):length(
      v)])
  psiNu[1:(length(v)/2-1)]<-Re(psiNu[(length(v)-1):(length(v)/2+1)])-1
      i*Im(psiNu[(length(v)-1):(length(v)/2+1)])
```

As before we use a loop to calculate $\hat{\nu}$ for different cut–off values. Since we have a separate cut–off `UNu` in the cases of "oracle" and "flat", the number of iterations is restricted to the selected cut–off `U` of the three parameters (in that way we save calculations, because $\hat{\nu}$ needs smaller cut–offs in general). The resulting $\hat{\nu}_U$ are stored in a matrix `nuHatCur`. Furthermore, we initiate some auxiliary variables and the `flatTopKernel`, as defined in (9).

```r
  if(mode=="oracle" | mode=="flat")
```

```
      cutOffIterations<-max(ceiling(U/(v[2]-v[1])),20)-10
   nuHatCur<-matrix(0,cutOffIterations,length(v))
   nuError2<--1
   if(mode=="oracle")
     nuError2Cur<-rep(0,cutOffIterations)
   if(mode=="PLS")
     objective<--1
   flatTopKernel<-function(x){(abs(x)<=0.05)+(abs(x)<1 & abs(x)>0.05)*(
       exp(-exp(-(abs(x)-0.05)^(-2))/(abs(x)-1)^2))}
   for(i in 1:cutOffIterations){
     if(mode=="fix")
       UCur<-UfixNu
     else
       UCur <- (v[2]-v[1])*(i+10)
     ...
   }
```

The first step in this loop is to calculate $\hat{\nu}_U$ with an inverse Fourier transform as in (8). Depending on the mode we use the final estimators $\hat{\sigma}, \hat{\gamma}$ and $\hat{\lambda}$ or the parameters per cut–off (since we have in the "PLS" mode a common cut–off). `xCur` denotes the grid on which the jump density is provided. It will be the same in every iteration step because it only depends on the grid `v` of `psiNu`.

```
     if(mode=="oracle" | mode=="flat" | mode=="fix"){
       lam<-lambdaHat
       fnuHat<-(psiNu-1i*gammaHat*v+lam+sigmaHat^2*v^{2}/2)*
           flatTopKernel(v/UCur)
     }
     else if(mode=="PLS"){
       lam<-lambdaHatCur[i]
       fnuHat<-(psiNu-1i*gammaHatCur[i]*v+lam+sigma2HatCur[i]*v^{2}/2)*
           flatTopKernel(v/UCur)
     }
     fxy_nu <- FT(v,fnuHat, noInverse=FALSE)
     xCur <- fxy_nu$u
     nuHatCur[i,] <- Re(fxy_nu$fy)
```

Up to now the condition $\hat{\nu}_U \geq 0$ is not ensured and needs a correction of $\hat{\nu}_U$. Referring to Belomestny and Reiß (2006b), we seek for a ξ such that the integral $\int_{\mathbb{R}} \max\{0, \hat{\nu}_U(x) - \xi\} \, \mathrm{d}x$ equals $\hat{\lambda}$ or $\hat{\lambda}_U$, respectively. We approximate the difference between the integral and `lam` as a function `eq` of `xi`, approximate its root, using the secants method, and redefine $\hat{\nu}_U$ as the corrected version.

```
     wNu<-rep(xCur[2]-xCur[1],length(xCur))
     wNu[1]<-0.5*wNu[1]
     wNu[length(wNu)]<-0.5*wNu[length(wNu)]

     if(lam>0){
       eq<-function(xi){sum(wNu*(pmax(0,nuHatCur[i,]-xi)))-lam}
       xi0<-0
       fxi0<-eq(xi0)
       xi1<-1
       fxi1<-eq(xi1)
       while(abs(fxi1)>0.001){
         xi2<-xi1-(xi1-xi0)/(fxi1-fxi0)*fxi1
         xi0<-xi1
         xi1<-max(0,xi2)
```

```
        fxi0<-fxi1
        fxi1<-eq(xi1)
    }
  }else
      xi2<-0
  nuHatCur[i,]<-pmax(0,nuHatCur[i,]-xi2)
```

Now we are in position to do the mode depending choice of $\hat{\nu}$. In the "PLS" mode we also select the other three parameters. In "oracle" we calculate for every UCur the L^2–distance to the true ν and pick the cut–off UNu with the smallest one. In "fix"–mode there is only one estimation.

```
  if(mode=="oracle"){
    nuTrue<-Nu(xCur)
    nuError2Cur[i]<-sum(wNu*(nuHatCur[i,]-nuTrue)^2)
    if(nuError2<0 | nuError2Cur[i]<nuError2){
      UNu<-UCur
      x<-xCur
      nuHat<-nuHatCur[i,]
      nuError2<-nuError2Cur[i]
    }
  }
  if(mode=="fix"){
    UNu<-UCur
    x<-xCur
    nuHat<-nuHatCur[i,]
    if(model!="real_data"){
      nuTrue<-Nu(xCur)
      nuError2<-sum(wNu*(nuHatCur[i,]-nuTrue)^2)
    }
  }
```

Still within the loop we deal with the "PLS" mode. Since it is driven by a least squares distance between the observations and the estimated model, we have to compute $\mathcal{O}$ from $\hat{\sigma}_U, \hat{\gamma}_U, \hat{\lambda}_U$ and $\hat{\nu}_U$. It is important that the martingale condition (1) holds for each U, otherwise the right–hand site of equation (3) can have a singularity. Therefore, we correct $\hat{\lambda}_U$ and $\hat{\nu}_U$ simultaneously. Remark that this is not done in the other modes (and actually cannot be done before the calculation of $\hat{\nu}$ is finished). The option pricing of (kE, opE) itself is described in detail in section B.3. From all available strikes kE we chose the ones next to the given observations sk and get (kHat, opHat).

```
  if(mode=="PLS"){
    beta<-(0.5*sigma2HatCur[i]+gammaHatCur[i])/(lambdaHatCur[i]-sum(
        wNu*exp(xCur)*nuHatCur[i,]))
    lambdaHatCur[i]<-beta*lambdaHatCur[i]
    nuHatCur[i,]<-beta*nuHatCur[i,]

    vE<--2^9/2+(0:(2^9-1))*2^9/(2^9-1)
    yE<-sapply(vE, function(v){ZetaHat(v, sigma2HatCur[i],
        gammaHatCur[i], lambdaHatCur[i], xCur, nuHatCur[i,])})
    fxy<-FT(vE,yE,FALSE)
    kE<-fxy$u
    opE<-Re(fxy$fy)
    kHat<-rep(0,length(sk))
    opHat<-rep(0,length(sk))
    for(j in (1:length(sk))){
      index<-which.min(abs(kE-sk[j]))
      kHat[j]<-kE[index]
```

```
        opHat[j]<-opE[index]
      }
      ...
    }
```

To penalize large second derivatives of ν, the second order differences of `nuHatCur` are provided in `nuDeriv`. The term to be minimized is given in `objectiveCur`. In every iteration step we check whether the new objective is smaller then the last best.

```
      nuDeriv<-diff(nuHatCur[i,],2)
      wE<-rep(xCur[2]-xCur[1],length(nuDeriv))
      wE[1]<-0.5*wE[1]
      wE[length(wE)]<-0.5*wE[length(wE)]
      alpha<-0 #1e-8
      objectiveCur<-sum((opHat-snop+pmax(0.,1-exp(sk-r*T)))^2)+alpha*
        sum(wE*(nuDeriv/(xCur[2]-xCur[1])^2)^2)
      if(!is.nan(objectiveCur) & (objective<0 | objectiveCur<objective
        )){
        U<-UCur
        UNu<-UCur
        sigmaHat<-sqrt(pmax(0,sigma2HatCur[i]))
        gammaHat<-gammaHatCur[i]
        lambdaHat<-lambdaHatCur[i]
        x<-xCur
        nuHat<-nuHatCur[i,]
        objective<-objectiveCur
        if(model!="real_data"){
          nuTrue<-Nu(xCur)
          nuError2<-sum(wNu*(nuHat-nuTrue)^2)
        }
      }
```

In the "flat" mode the choice of U_ν is made at the end of the loop. We approximate the L^2–norm of the derivative of the map $U \mapsto \hat{\nu}_U$ and select its minimum.

```
  if(mode=="flat"){
    derivativU<-diff(nuHatCur)/(v[2]-v[1])
    derivativL2<-rep(0,length(derivativU[,1]))
    for(i in (1:length(derivativU[,1])))
      derivativL2[i]<-sum(wNu*derivativU[i,]^2)
    i<-which.min(derivativL2)
    UNu<-(v[2]-v[1])*(i+10)
    x<-xCur
    nuHat<-nuHatCur[i,]
    if(model!="real_data"){
      nuTrue<-Nu(xCur)
      nuError2<-sum(wNu*(nuHat-nuTrue)^2)
    }
  }
```

Now we have all estimators and can return the results.

B.3   Option pricing from Lévy triplet

Given values of σ, γ, λ and ν we want to provide the $\mathcal{O}$–function. This happens in two situations of the script. On the one hand we simulate observations from a specific model and apply the

calibration to the noised data and on the other hand, we have to calculate option prices from the estimated parameters in the least squares method. In (Cont and Tankov, 2004b, p. 361 et seq.) a good description of this option pricing can be found.

We define the function

$$\zeta(v) := e^{ivrT} \frac{\phi_T(v-i) - 1}{iv(1 + iv)}$$

which is implemented as `Zeta(v)` and `ZetaHat(v, sigma2Hat, gammaHat, lambdaHat, x, nuHat)`, respectively. Then $\mathcal{O}$ as function of log–strike k is obtained by inverse Fourier transform of ζ. Hence, we evaluate ζ on a discrete grid `x` to get pairs of log–strikes and option prices (`k`, `op`). Because of the FFT algorithm a finer grid `x` leads to a larger limits of `k` and the other way around a higher range of `x` implies a smaller mesh of `k`.

```
A<-2^10
M<-12
N<-2^M
l<-0:(N-1)
Delta<-A/(N-1)
x<--A/2+l*Delta
y<-Zeta(x)
fxy<-FT(x,y,FALSE)
k<-fxy$u
op<-Re(fxy$fy)+pmax(0.,1-exp(k-r*T))
```

For simulations we noise and sample all pairs (`k`,`op`) and apply the calibration procedure in every Monte Carlo iteration step. Here the sampling is done according to a normal distribution centered at at–the–money strikes. Depending on `design`, the strikes a distributed deterministic or random.

```
for(mi in 1:monteCarlo){
   nop<-op+rnorm(length(op),0,(noiseLevel*abs(Re(fxy$fy))))

   M1<-round(length(op)/2)-which.min(abs(op[1:round(length(op)/2)
      ]-10^(-6)))
   M2<-which.max(op)+M1
   M1<-which.max(op)-M1

   if(design=="random"){
      prob<-exp(-(k[M2:M1]-r*T)^2)
      prob<-prob/sum(prob)
      ind_sub<-sort(sample(M2:M1,sampleSize,prob=prob))
      sk<-k[ind_sub]
      snop<-nop[ind_sub]
   }else if(design=="deterministic"){
      quantils<-qnorm(c(1:sampleSize)/(sampleSize+1),0,2^(-0.5))
      ind_sub<-sapply(quantils,function(x){which.min(abs(k-r*T-x))})
      sk<-k[ind_sub]
      snop<-nop[ind_sub]
   }
   snop<-snop+pmax(0.,1-exp(sk-r*T))
   ...
}
```

References

Aït-Sahalia, Y. and J. Jacod (2009). Estimating the degree of activity of jumps in high frequency data. *Ann. Statist. 37*(5A), 2202 – 2244.

Asmussen, S., F. Avram, and M. R. Pistorius (2004). Russian and American put options under exponential phase-type Lévy models. *Stochastic Processes and their Applications 109*(1), 79 – 111.

Barndorff-Nielsen, O. E. (1998). Processes of normal inverse gaussian type. *Finance Stoch 2*, 41–68.

Belomestny, D. and M. Reiß (2006a). Spectral calibration of exponential Lévy models. *Finance Stoch 10*, 449–474.

Belomestny, D. and M. Reiß (2006b). Spectral calibration of exponential Lévy models [2]. SFB 649 Discussion Paper 2006-035, Sonderforschungsbereich 649, Humboldt Universität zu Berlin, Germany. Available at http://sfb649.wiwi.hu-berlin.de/papers/pdf/SFB649DP2006-035.pdf.

Carr, P., H. Geman, D. B. Madan, and M. Yor (2002). The fine structure of asset returns: An empirical investigation. *J. Bus. 75*(2), 305–332.

Carr, P. and D. B. Madan (1999). Option valuation using the fast Fourier transform. *J. Comput. Finance 2*, 61–73.

Cont, R. (2006). Model uncertainty and its impact on the pricing of derivative instruments. *Math. Finance 16*(3), 519–547.

Cont, R. and P. Tankov (2004a). *Financial modelling with jump processes*. Chapman & Hall / CRC Press.

Cont, R. and P. Tankov (2004b). Non-parametric calibration of jump-diffusion option pricing models. *J. Comput. Finance 7*(3), 1–49.

Cont, R. and E. Voltchkova (2005). Integro-differential equations for option prices in exponential Lévy models. *Finance and Stochastics 9*, 299–325.

Eberlein, E., U. Keller, and K. Prause (1998). New insights into smile, mispricing and value at risk: The hyperbolic model. *J. Bus. 71*, 371–406.

Figueroa-López, J. (2011). Sieve-based confidence intervals and bands for Lévy densities. *Bernoulli 17*(2), 643–670.

Härdle, W. (1990). *Applied nonparametric regression*. Cambridge University Press.

Ivanov, R. V. (2007). On the Pricing of American Options in Exponential Lévy Markets. *Journal of Applied Probability 44*(2), 409–419.

Madan, D. B., P. P. Carr, and E. C. Chang (1998). The variance gamma process and option pricing. *Europ. Finance Rev. 2*(1), 79–105.

Madan, D. B. and E. Seneta (1990). The variance gamma (VG) model for share market returns. *J. Bus. 63*, 511–524.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *J. Finan. Econ. 3*(1-2), 125 – 144.

Söhl, J. (2012). Confidence sets in nonparametric calibration of exponential Lévy models. SFB 649 Discussion Paper 2012-012, Sonderforschungsbereich 649, Humboldt Universität zu Berlin, Germany. Available at http://sfb649.wiwi.hu-berlin.de/papers/pdf/SFB649DP2012-012.pdf.

Trabs, M. (2011). Calibration of self-decomposable Lévy models. SFB 649 Discussion Paper 2011-073, Sonderforschungsbereich 649, Humboldt Universität zu Berlin, Germany. Available at http://sfb649.wiwi.hu-berlin.de/papers/pdf/SFB649DP2011-073.pdf.